

Grid Policy Administrator with MDP-Based Resource Management Scheme

Sang-Il Lee, Youngjoo Han, Hyewon Song, Changhee Han, Chan-Hyun Youn

*School of Engineering, Information and Communications University
119, Munjiro, Yuseong-gu, Daejeon, 305-732, Korea
{vlsivlsi, yjhan, hwsong, changhee, chyoun}@icu.ac.kr*

Abstract— Grid computing enables the virtualization of distributed computing and data resources such as processing, network bandwidth and storage capacity to create a single system image. The complexity of resource management in Grid significantly increases as the grid computing grows. To solve the problem we propose grid policy administrator with MDP-based resource management scheme. The grid policy administrator has a QoS-constrained grid resource management architecture based on Quorum and MDP-based policy decision scheme for considering management cost and QoS guarantee. Available resource quorums are generated by preplanned policy decision scheme with various SLAs, grid resource and network resource information in advance. The results show that our proposed scheme guarantees flexible QoS for grid applications.

Keywords

Policy based Resource Management, QoS Quorum, MDP-Based Resource Management, and Overlay Grid Network

I. INTRODUCTION

Grid computing brings together all heterogeneous resources and allocates them efficiently to applications. However, the complexity of resource management for QoS guarantee in Grid significantly increases as the grid computing grows. There are some approaches that try to solve the problem. One of the promising is policy based resource management system that is suitable to the complex management environments.

In this paper, we propose grid policy administrator with MDP-based resource management scheme. The grid policy administrator has a QoS-constrained grid resource management architecture based on Quorum and MDP-based policy decision scheme for considering management cost and QoS guarantee. The management architecture has two phase operations. The first complex decision procedure in these operations runs in proactive way to reduce brokering time. Then the second reactive procedure triggered by on-demand resource requests is served in time. Quorum systems are well known tools for ensuring the consistency of resource information as well as reliable QoS guarantee. Its useful properties such as intersection and mutual exclusion are used to make a large set of distributed heterogeneous resources.

Available resource quorums are generated by preplanned policy decision scheme with various SLAs, grid resource and network resource information in advance. When a request arrives, those available resources are mapped into QoS quorums which satisfy QoS requirements. To evaluate the preplanned policy decision scheme with policy enforcement into network we run virtual heart simulation in our experiments. The results show that our proposed scheme guarantees QoS for grid applications.

II. RELATED WORKS

The traditional Grid computing (Computational, Data, Access) [1][2][3] is more specialized, and consists of various and complex grid services. For the reliable management of physically distributed and heterogeneous resources, policy-based management [6] has been suggested. The connection between lower network and the resource management system is mandatory for such a grid policy-based resource management system. It is requisite to make the resource management system abstract because it is hard to manage heterogeneous lower management system managed by local policy of different subject like a grid plane.

The research in Dons Su Nam et al. [4] suggested policy-based resource management system architecture based on grid SLA and abstract architecture of grid resource management. For guarantying QoS, it defines the quorum of QoS parameter; the set based on guarantying QoS and suggested the resource allocation mechanism via well-defined algorithm. And the research in Byung et al. [5] suggested the scheme for the reset of quorum for guarantying QoS, coping with variation of resource status based on evaluation of quorum in time domain. However, in QoS part of grid, guarantying QoS is essential element, the research in Ricardo Neisse et al. [9] suggested the hierarchical architecture mapping grid resource management policy into the policy of network.

In [7], it suggested the way of guarantying QoS in network based on active network for QoS and policy-

based resource management for managing active network. However, it has considered only network policy and it is not related with the grid policy. And also Yang et al. [8] prescribes the core of grid as a RoD(Resource on Demand) and suggested the probability of variable QoS setting by active network based on policy-based network for guarantying QoS effectively. From those researches, the guarantying QoS demands on network QoS and under the control of each domain administer for heterogeneous resource management, policy-based management system is suggested as an effective structure. This paper reviews the results of a series of previously study aimed at conventional resource management schemes [6][7][9], Policy based Management, QoS guarantees, Service Level Agreement, Quorum based resource abstraction in Grid. Therefore, we study a solution suitable for reliable, flexible resource management under incentive to various types of individual applications to contribute to the success of a goal.

III. GRID POLICY ADMINISTRATOR WITH MDP-BASED RESOURCE MANAGEMENT SCHEME

Although there are architectures for the resource management in Grid, no existing system can consider whole resource management features. That is, all of grid management system focuses on its own purpose.

Policy based management (PBM) system [6][7] is originated from network management groups to reduce the complexity of management in terms of QoS guarantees. It is suitable to complex management environment such as large scale heterogeneous resource management in Grids. Quorum is a collection of the elements which always intersects. It has been used to maintain the consistency of replicated data, mutual exclusion. In PQRM the intersection property of Quorum is used to make complex resource elements abstracted Grid resource which satisfies various QoS requirements. The main result of PQRM is that QoS-constraint policy has an advanced method to satisfy QoS requirements requested by applications or services [8].

A. Architecture of Policy mapping in PQRM

The Quorum has subset of resource pool which is discovered by GIS (Grid Information service) such as MDS (Meta-computing Directory Service) in Globus & a network information service in PBNM (Policy based Network Management). QoS Quorum is generated in Quorum Generator module with QoS requirements, User information and resource information. In conventional architectures the optimization of resource allocation has been based on grid middleware level policy. To satisfy grid level goals those lower level components support high level goals. Similarly, network & system policies support

grid policy to achieve grid goal. Thus, we add Policy Engine in PQRM to enforce grid policy to network or system policies.

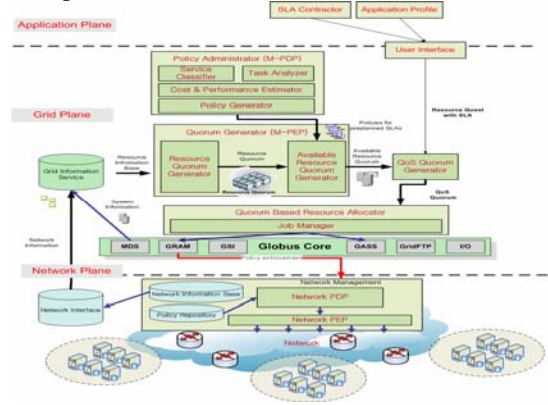


Figure 1 Policy Administrator in Grid Resource Management

B. Quorum Models in Grid

1) System Resource Model

Basically, layer 7 resources such as server, storage, clusters so as on are modeled as system resources in Grid. The system resource has various attributes such as CPU, memory, I/O, storage... etc [4].

$$\mathbf{SR} = \{sr_1, sr_2, \dots\} \quad (1)$$

2) Network Resource Model

End-to-end network is modeled as network resource. L3/L2/L1 resources such as routers, switches, hubs and so on have been treated as grid network recently [4].

$$\mathbf{NR} = \{nr_{12}, \dots, nr_{ij}\} \quad (2)$$

3) Resource Quorum Model

Def. 1) Resource Quorum is a set of resources that a broker has their system & network information in resource universe.

$$\mathbf{RQ} = \{r_1, r_2, \dots, r_M\} = \{<sr_i, nr_{ij} > | i \neq j \quad i, j = 1, \dots, n\}$$

$$= \mathbf{SR} \cdot \mathbf{I} + \mathbf{NR} = \begin{pmatrix} sr_1 & nr_{12} & \cdots & nr_{1n} \\ nr_{21} & sr_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & nr_{(n-1)n} \\ nr_{n1} & \cdots & nr_{n(n-1)} & sr_n \end{pmatrix} \quad (3)$$

,where \mathbf{RQ} is a resource quorum and r_j (the element of \mathbf{RQ}) is a tuple of system resource sr_j and network resource nr_j .

Lemma1. Resource Quorum is a subset of resource universe.

$$\mathbf{RQ} \subseteq \mathbf{U} \quad (4)$$

Proof: a grid resource broker only knows some of GISs in global environments because of scalability problem. And any resource R_i of resource quorum \mathbf{RQ} comes from grid resource universe \mathbf{U} . If a resource joins in or leaves from grid resource universe \mathbf{U} , a broker knows checking the intersection

part of resource quorum at different time. Thus, the consistency between resource information a broker has and resource information in grid resource universe is maintained. Consequently, any resource R_j of resource quorum \mathbf{RQ} exists in grid resource universe \mathbf{U} .

4) Available Resource Quorum Model

a) Resource allocation cost function

We can define the SLA type using QoS factors. When these parameters are given by a Network Information Base and a System Information Base, we can define the network resource allocation cost (C_{Net}) and system resource allocation cost (C_{Sys}) according to these SLA types, which are different as follows: allocation cost (C_{Net}) and system resource allocation cost (C_{Sys})

,where l means the QoS factors using for the SLA, that is, L is the number of QoS factors which use in the SLA process. a and b is unit cost factor. Using these equations, we can define the resource allocation cost function as follows:

$$C = \gamma \cdot C_{Net} + (1 - \gamma) \cdot C_{Sys}, \quad 0 < \gamma \leq 1 \quad (6)$$

,where γ is a weighting factor.

b) Optimality Function for Policy Decision

Assume that there are M resources. When a job is assigned in resource i , the resource allocation cost function is defined as a following equation using Eq. (6).

$$C_i(t) = \gamma \cdot C_{ij}^{Net}(t) + (1 - \gamma) \cdot C_i^{Sys}(t) \quad (7)$$

,where $0 < \gamma < 1$ and $i, j = 1, 2, \dots, M$. γ is a weighting factor, and l means the QoS factors using for the SLA. This cost is given by sum of network resource allocation cost and system resource allocation cost.

$$C_i(t) = \gamma \cdot C_{ij}^{Net}(t) + (1 - \gamma) \cdot C_i^{Sys}(t) \quad (8)$$

$$C_i(t) = \gamma \cdot \sum_{l=1}^L a_{ij}^l \cdot C_{nr}(nr_{ij}^l(t)) + (1 - \gamma) \cdot \sum_{l=1}^L b_{ij}^l \cdot C_{sr}(sr_{ij}^l(t)) \quad (9)$$

When the job is submitted to Grid resources, resource i is assigned to this job with the probability p_i . At this point we can (a) decide to go this resource or (b) choose to hold the resource until resource is available. If the latter decision is made, we assume the system periodically returns to this resource always with the same probability distribution p_i ($i = 1, 2, \dots, M$) in any SLA 1. However, until the next resource assignment is made, the resource has to be held and incur a fixed finite cost like a penalty cost $C_{a,i}(t)$ according to time in resource i .

$$C_{a,i}(t) = c_{fr} \cdot C_{fr,i}(t), \quad i = 1, 2, \dots, M \quad (10)$$

c_{fr} is unit cost factor given as constant. $C_{fr,i}(t)$ means the cost by historical failure rate in resource i and is defined as follows:

$$C_{fr,i}(t) = C_{fr,i}^{sys}(t) + C_{fr,i}^{net}(t) \quad (11)$$

$C_{fr,i}^{sys}(t)$ and $C_{fr,i}^{net}(t)$ mean the cost by historical system failure rate and the cost by historical network failure rate in resource i , respectively. Also, these are defined as follows:

$$C_{fr,i}^{sys}(t) = H_i^{sys}(t), \quad i = 1, 2, \dots, M \quad (12)$$

$$C_{fr,i}^{net}(t) = \sum_{j \neq i} H_{ij}^{net}(t), \quad i, j = 1, 2, \dots, M$$

, where $H_i^{sys}(t)$ means the historical system failure rate in resource i at time t . $H_{ij}^{net}(t)$ means the historical network failure rate between source resource i and destination resource d at time t .

Clearly, the tradeoff in this problem is between the accumulation of penalty costs and the hope of a low-cost resource allocation. In this paper, Markov decision process [10] is used for resource allocation function. When i is first resource to assign to a quorum with $i \in \{1, 2, \dots, M\}$, control actions is defined as follows:

$$x(i) = \begin{cases} 1 & \text{if the job given with SLA 1 is assigned in resource } i \\ 0 & \text{if the job given with SLA 1 is not assigned in resource } i \end{cases} \quad (13)$$

Using Eq. (9), (10) and (11), we can define the optimality function for a policy decision with Markov decision process when α is a given discount factor, $0 < \alpha < 1$.

$$F(i, t) = \min \left[x(i)C_i(t) + (1 - x(i))C_{a,i}(t) + (1 - x(i))\alpha \sum_{j=1}^M p_j F(j, t) \right] \\ = \min \left[x(i) \left\{ \gamma \cdot \sum_{l=1}^L a_{ij}^l \cdot C_{nr}(nr_{ij}^l(t)) + (1 - \gamma) \cdot \sum_{l=1}^L b_{ij}^l \cdot C_{sr}(sr_{ij}^l(t)) \right\} \right. \\ \left. + (1 - x(i)) \left\{ C_{a,i}(t) + \alpha \sum_{j=1}^M p_j F(j, t) \right\} \right] \quad (14)$$

Here, this function can be represent as follows:

$$F(i, t) = \min \left[C_i(t), C_{a,i}(t) + \alpha \sum_{j=1}^M p_j F(j, t) \right] \\ = \min \left[\gamma \cdot \sum_{l=1}^L a_{ij}^l \cdot C_{nr}(nr_{ij}^l(t)) + (1 - \gamma) \cdot \sum_{l=1}^L b_{ij}^l \cdot C_{sr}(sr_{ij}^l(t)), C_{a,i}(t) + \alpha \sum_{j=1}^M p_j F(j, t) \right] \quad (15)$$

Using this function we can derive Lemma 1.

Lemma2. If $C_i(t) < C_{a,i}(t) + \alpha \sum_{j=1}^M p_j F(j, t)$, the way to minimize $F(i, t)$ is by choosing the control action $x(i) = 1$

Proof: If $C_i(t) < C_{a,i}(t) + \alpha \sum_{j=1}^M p_j F(j, t)$, $C_i(t)$ is selected, that is, the job is assigned in resource i as shown in Eq. (6). Thus, in order to minimize $F(i, t)$, the control action, $x(i) = 1$ is chosen. When the value of $C_i(t)$ is rearranged to an ascending series $C_k(t)$, it follows that there is some critical value k^* such that $C_{k^*}(t)$ is sufficiently small because C_k is increasing in k , that is

$$C_{k^*}(t) < C_{a,i}(t) + \alpha \sum_{j=1}^M p_j F(j, t) \quad (16)$$

We immediately see that the optimal policy is to hold the resource until we assign resource where $k \leq k^*$, where k^* is defined by Eq. (16). The

optimal control action at state $k=1, 2, \dots, M$, denoted by $x^*(k)$, is therefore

$$x^*(i) = \begin{cases} 1 & \text{if } k \leq k^* \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

However, we still need to evaluate $F(k, t)$, $k=1, 2, \dots, M$, in order to apply this optimal policy, since these values are needed for the determination of the critical index k^* in Eq. (16). For this, we derive the decision function and the rule as following theorem.

Theorem. When the decision function (refer to Appendix) is defined as Eq. (17),

$$G(n, t) = \frac{C_{a,i}(t) \sum_{i=n+1}^M p_i + \sum_{i=1}^n C_i(t) p_i}{1 - \alpha \sum_{i=n+1}^M p_i} \quad (18)$$

If $n \leq i^*$, any resource such that $i \leq n$, satisfies the optimal policy.

Proof: In order to derive a kind of threshold function which determines whether the resource is assigned in a quorum or not, we define the optimal policy β^* . For this function, used variable I is rearranged by ascending series. This means the policy that allocates any resource such that $i \leq i^*$ from Eq. (17), $i^* \in \{1, 2, \dots, M\}$. When a policy β_n allocates any resource such that $i \leq n$. Clearly, $\beta^* = \beta_n$ when $n = i^*$.

Let R denote the random number of rejected resource allocations under policy β_n . Then, conditioned on the value of R , the total expected cost under β_n is

$$E_{\beta_n} \left[\sum_{q=0}^{\infty} \alpha^q \cdot C(X_q, x_q) \mid R=r \right] = \left[C_{a,i}(t) \cdot (1 + \alpha + \dots + \alpha^{r-1}) \right] + \alpha^r E[C_i \mid i \leq n] \quad (19)$$

Here, the first r terms in brackets on the right-hand side are the accumulated discounted penalty costs for the r rejected resource allocation. The last term is the expected cost incurred by the assigned resource, conditioned on our policy that only a resource $i \leq n$ is accepted. As shown in Appendix, we can derive Eq. (18) from Eq.(9). From this $G(n, t)$, we can determine n , and then, using Eq. (7) and Lemma 1, any resource such that $i \leq n$, satisfies the optimal policy.

Using this theorem, we can obtain the available resource set and then we can create the Resource Allocation Table as shown in Figure 2.

5) QoS Quorum Model

a) Service Level Agreement Model

$$\mathbf{S} = \{s_1, \dots, s_s\} \quad (20)$$

, where \mathbf{S} is the system service level objective and s_j represents the j_{th} system resource SLO vector

$$\mathbf{N} = \{n_1, \dots, n_s\} \quad (21)$$

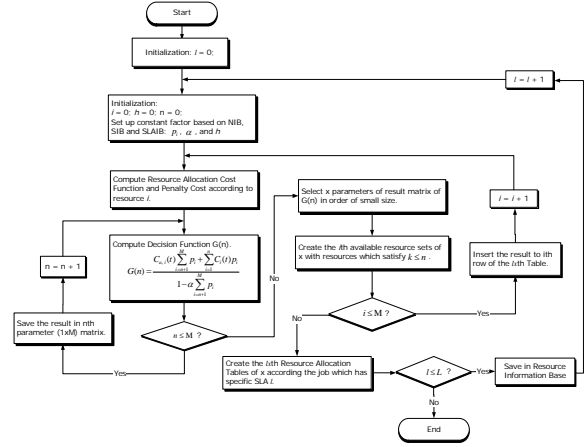


Figure 2 Policy Decision Algorithm

, where \mathbf{N} is the system service level objective and n_j represents the j_{th} network resource SLO vector

$$c(o_i) \in \{1, 0\} \quad (22)$$

, where $c(o_i)$ is the care of the i_{th} SLO elements. If $c(o_i)$ is "1", SLA cares the i_{th} SLO, Otherwise "0". For instance, $c(s_2)=1$ means that SLA cares the second system QoS requirement.

$$f(o_i) \in [0, \infty) \quad (23)$$

, where $f(o_i)$ is the function that returns requirement values.

b) System QoS Vector

$$\mathbf{SV}_i = \{ \bigcup_{vj} sr_j \mid sr_{ij} \geq f(s_i) \quad j=1, \dots, n \} \quad (24)$$

Where, \mathbf{sv}_i is the i_{th} system QoS vector that represents a set of system resources which satisfy the i_{th} system QoS requirement in SLA.

$$\mathbf{SQ} = \bigcap_{\forall i, c(s_i) \neq 0} \mathbf{SV}_i = \{ sr_1, \dots \} = \begin{Bmatrix} sr_{11} & \dots & sr_{1n} \\ \vdots & \ddots & \vdots \\ sr_{1m} & \dots & sr_{1m} \end{Bmatrix} \quad (25)$$

Where, \mathbf{SQ} is the system QoS quorum that represents the intersection set of the \mathbf{SV} s with $c(s) \neq 0$, it means that \mathbf{SQ} are dependant on SLA.

c) Network QoS Vector

$$\mathbf{NV}_i = \{ \bigcup_{vj} nr_j \mid nr_{ij} \geq f(n_i) \quad j=1, \dots, n \} \quad (26)$$

Where, \mathbf{nv}_i is the i_{th} network QoS vector that represents a set of network resources which can satisfy the i_{th} network QoS requirement in SLA.

$$\mathbf{NQ} = \bigcap_{\forall i, c(n_i) \neq 0} \mathbf{NV}_i = \{ nr_1, \dots \} = \begin{Bmatrix} nr_{11} & \dots & nr_{1n} \\ \vdots & \ddots & \vdots \\ nr_{1m} & \dots & nr_{1m} \end{Bmatrix} \quad (27)$$

Where, \mathbf{NQ} is the system QoS Quorum that represents the intersection set of the \mathbf{NV} s with $c(s) \neq 0$.

d) QoS Quorum

Def. 1) QoS Quorum is a set of resources that satisfy QoS requirement in SLA.

$$\mathbf{Q} = \{ \langle sr_i, nr_{ij} \rangle | i \neq j \quad i, j = 1, \dots, m \}$$

$$\{ sr_{v,k,i} \geq c(s_k) \times f(s_k), nr_{ij}^k \geq c(n_k) \times f(n_k) \}$$

$$= \mathbf{SQ} \cdot \mathbf{I} + \mathbf{NQ} = \begin{pmatrix} sr_1 & nr_{12} & \dots & nr_{1m} \\ nr_{21} & sr_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & nr_{(m-1)m} \\ nr_{m1} & \dots & nr_{m(m-1)} & sr_m \end{pmatrix} \quad (28)$$

, where \mathbf{Q} is the QoS guaranteed resource set, which is a subset of the resource quorum \mathbf{RQ} , and i and j are elements in \mathbf{Q} . sr_i is a system resource that represents a QoS guaranteed computing node. nr_{ij} is a network resource that represents a QoS guaranteed communication network between system resource i and j . Then we can find $i^{Min_QoS} \leq i$. Thus $i^{Min_QoS} \leq i \leq i^*$ condition is applied to form QoS Quorum for QoS guarantee and management cost optimization.

C. Quorum based Resource Management Operation

Quorum creation consists of two step operation, proactive and reactive operation.

1) Proactive Quorum Operations

Proactive resource management maintains the performance of management system within the boundary of anticipating critical changes in the management system states, pre-planned resource allocation appropriate to those changes. In grid resource management system gathering resource information & manipulation take much time. Thus, before grid customers request the management system maintain the resource set as a quorum.

2) Reactive Quorum Operations

Reactive operation is on demand process of resource allocation to grid user's jobs. A grid customer submits a job to grid system with user information, SLA, application information so as on. Then, policy decision engine has a decision role to admit this request through policy enforcement process.

IV. SIMULATION AND RESULTS

A. Numerical Analysis

1) Algorithm for Policy Decision Propose

In this section, we will evaluate the complexity of the policy decision process by comparing existing model to our proposed model. Previous model is an integrated one which does not divide the processes between available resource Quorum and QoS Quorum creation. Proposed model divides these two steps using proactive and reactive process. Table 1 and

Table 2 Overlay Model (Proposed Model) show the pseudo code of the policy decision process between integrated and overlay model respectively.

Table 1 Integrated Model for Policy Decision

```

set up constant factor based on NIB, SIB, SLAIB;
for(i; i<=M; i++){compute resource allocation cost fu
nction;
  compute penalty cost according to resource i;
  for(n; n<=M; n++){ compute Decision Function: G
(n);
    save the result in nth parameter(1xM) matrix;
  }
  select x parameters of result matrix of G(n);
  create the ith available resource set; }
create the 1xth resource allocation table of ~;
save in Resource Information Base;

```

Table 2 Overlay Model (Proposed Model)

```

for(l; l<=L; l++){ set up constant factor based on NIB,
SIB, SLAIB;
  for(i; i<=M; i++){compute resource allocation cost f
unction;
    compute penalty cost according to resource i;
    for(n; n<=M; n++){ compute Decision Functio
n: G(n);
      save the result in nth parameter(1xM) matrix; }
      select x parameters of result matrix of G(n);
      create the ith available resource set; }
      create the 1xth resource allocation table of ~;
      save in Resource Information Base; }

```

2) Complexity of Policy Decision Process

Based on Table 1 and Table 2, we can calculate the complexity using Big O notation. Figure 3 shows the complexity between two models. Both have $O(n^2)$. Our proposed model has $O(K \cdot n^2)$ and the integrated model has $O(R \cdot n^2)$ where K is the number of SLA types and R is the number of resource request within a given time. Thus the proposed model is more beneficial when R is bigger than K .

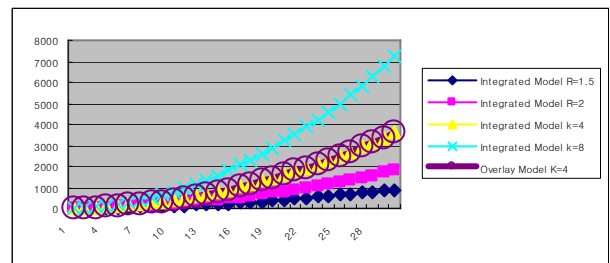


Figure 3 Comparison the Complexity between Integrated Model and Overlay Model

B. Experimental Result

1) Environment for Experiment

We established the Grid testbed for collaborating among multi-domain environments. We use Linux-based Globus platform for Grid application.

Table 3 Node Specification on the Grid Testbed

Node	Host name	Grid platform
S1	physio5.icu.ac.kr	MPICH-G2 / Globus2.4.3
S2	physio6.icu.ac.kr	MPICH-G2 / Globus2.4.3
S3	physio7.icu.ac.kr	MPICH-G2 / Globus2.4.3
S4	pPhysio8.icu.ac.kr	MPICH-G2 / Globus2.4.3
S5	physio10.icu.ac.kr	MPICH-G2 / Globus2.4.3
S7	physio11.icu.ac.kr	MPICH-G2 / Globus2.4.3
S8	physio12.icu.ac.kr	MPICH-G2 / Globus2.4.3
S9	physio13.icu.ac.kr	MPICH-G2 / Globus2.4.3
S10	physio14.icu.ac.kr	MPICH-G2 / Globus2.4.3

2) Applications for Grid Simulation

On this section, we introduce the parallel application that communicates among each sub-job. End-to-end communication quality of service in case of linked chains is more important issue because it is influenced on the entire performance of the application execution.

Blood flow in the sac of the Korean artificial heart (KAH) is numerically simulated by finite element method. Distributed computing algorithm is employed to compute using Globus-based MPI programming. Figure 4 shows the Boundary condition of the KAH.

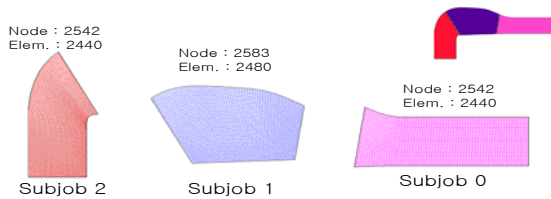


Figure 4 Finite Element Model of the KAH

We experiment the KAH with three domains and use five machines on the Table 3 Node Specification on the Grid Testbed. In order to get the QoS Quorum, we should set up minimum performance for execution. Table 4 shows the resource requirement for executing the KAH.

We experiment with offered load at system on the time step 50 and network on the time step 100 and both on the time step 140. Figure 5 and Figure 6 show the execution time of each step for processing the

KAH when not applying any Grid policy and preplanned decision policy respectively.

Table 4 QoS requirements for KAH

Node	HostName	CPU ratio (%)	BW(Mbps)
S1	physio5.icu.ac.kr	75%	1
S3	physio7.icu.ac.kr	30%	1
S8	physio11.icu.ac.kr	30%	1
S9	physio12.icu.ac.kr	35%	1
S10	physio13.icu.ac.kr	30%	1

On the Figure 5, the execution time was largely fluctuated by the offered load from the external environment. On the Figure 6, we can notice the stability of the execution time when applying the preplanned decision policy.

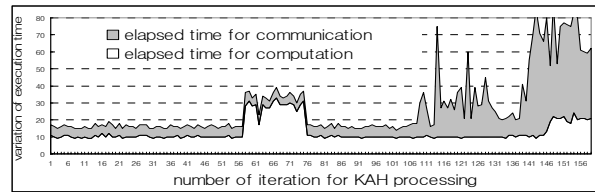


Figure 5 Execution time of Each Step (Non Policy)

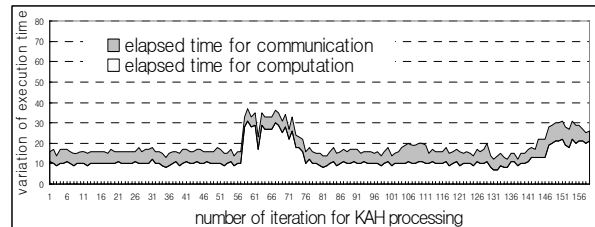


Figure 6 Execution time of Each Step (Proposed)

Figure 7 shows that execution time comparison among three different policies. Through the proposed scheme, total execution time is decreased by decreasing and communication time compared to other resource allocation policy.

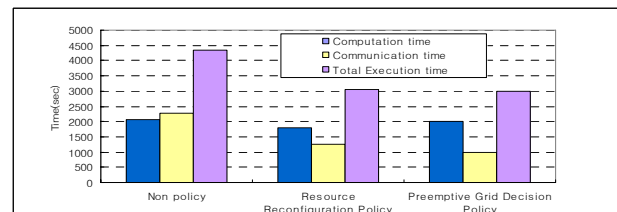


Figure 7 Comparisons for Total Execution Time among Three Different Policies

V. CONCLUSION

In this paper, we proposed policy quorum based resource management architecture and preplanned grid policy decision scheme for QoS guarantee and reduce of complexity. First of all, in the proposed resource management architecture there are two phase operation to create QoS quorum. Then, experimental results show that our approaches guarantees QoS by policy based resource management system and reduce the complexity of management using preplanned SLAs. In the result of VH simulation our approaches decreasing total execution time reducing communication time by network policy enforcement from grid resource management system.

VI. ACKNOWLEDGEMENT

This research was supported by MIC (Ministry of Information and Communication) of Korean government.

VII. REFERENCES

- [1] Foster, I. et al. "The Grid: Blueprint for a New Computing Infrastructure". Morgan Kaufmann, 1999.
- [2] Foster, I. et al. "The Anatomy of the Grid:Enabling Scalable Virtual Organizations". Intl J. Supercomputer Applications, 2001.
- [3] Czajkowski, K. et al. "Grid Information Services for Distributed Resource Sharing", 2001.
- [4] Dong Su Nam et al. "QoS-constraint Quorum Management scheme for a Grid-based Electrocardiogram", LNCS 3043 pp.352-359, 2004
- [5] Byung Sang Kim et al. "Resource Reconfiguration Policy based on Temporal Quorum Status Estimation Scheme in Computational Grids" ICOIN 2004, 699-707
- [6] Byung Sang Kim et al. "Policy Quorum based Resource Management Architecture in Grids", IJCSNS2005, Vol5, No 8
- [7] I. Liabotis et al. "Policy-Based Resource Management for Application Level Active Networks", LANOMS 2001
- [8] Kun Yang et al. "Efficient Resource on-Demand in Grid Computing", ICCT 2003. Volume 2, 9-11 April 2003.
- [9] Ricardo Neisse et al. "An Hierarchical Policy-Based Architecture for Integrated Management of Grids and Networks," policy, p. 103 (POLICY'04), 2004.
- [10] Discrete Event Systems: Modeling and Performance Analysis